# Large Scale & Distributed Optimization

**Angelia Nedić**

January 19, 2022

ASU
ARIZONA STATE UNIVERSITY

contact: *Angelia.Nedich@asu.edu; angelianedich@gmail.com*

# Lecture 3:

# Fast Distributed Algorithms


In memory of Wilbur Wei Shi

# Recall Consensus-Based Method

- The agents communicate over undirected connected graphs, and $N_{it}$ is the set of neighbors of agent $i$
- At time $t$, every agent $i$ sends $x_i(t)$ to its neighbors $j \in N_i$, and receives $x_j(t)$ from them; then, every agent updates (AN and A. Ozdaglar 2009)

$$x_i(t+1) = \underbrace{\sum_{j=1}^{m} W_{ij}(t)x_j(t)}_{\text{consensus}} - \alpha_t \nabla f_i(x_i(t)) \quad \text{where } \alpha_t > 0 \text{ is a stepsize}$$

- Assuming that the problem has a solution and some other conditions, with stepsize

$$\sum_{t=0}^{\infty} \alpha_t = \infty, \qquad \sum_{t=0}^{\infty} \alpha_t^2 < \infty, \tag{1}$$

each agent decision $x_i(t)$ converges to a common optimal solution $x^*$ of the system problem,

$$\lim_{t\to\infty} x_i(t) = x^* \qquad \text{for all } i,$$

where $x^*$ is a minimizer of $\sum_{j=1}^{m} f_j(x)$ over $x \in \mathbb{R}^n$.

- Convergence rate at best is in the order of $\log t/t$ (for strongly convex objective function). The rate is due to the use of the stepsize satisfying (1)
- The matrices $W(t)$ are doubly stochastic

# Yet Another Issue

- The consensus-type algorithms discussed thus far will not produce convergent iterates when **a fixed stepsize** is used
- Brought to our attention in the work of Wilbur (Wei) Shi, Ling, Wu, and Yin (EXTRA) 2014, 2015
- Suppose the graph is static and the stepsize is fixed to $\alpha > 0$
- Suppose that the agent sequences $\{x_i(t)\}$ are all converging to some point $x^*$

$$x_i(t+1) = \sum_{j=1}^{m} w_{ij} x_j(t) - \alpha \nabla f_i(x_i(t))$$

$$x^* = x^* - \alpha \nabla f_i(x^*) \implies \nabla f_i(x^*) = 0 \qquad \text{for all } i \in [m]$$

- This can happen if all $f_i$ have a common minimizer $x^*$!

- Implications:
  - In general, the method with a fixed stepsize will not produce convergent iterates
  - Agreement among the agents will be reached in a long run
  - The agents' iterates will be trapped in some region where they will keep "bouncing" within
  - The method cannot be accelerated
- Fact: When $f : \mathbb{R}^n \to \mathbb{R}$ is strongly convex and has Lipschitz continuous gradients, i.e., for some $\mu > 0$ and $L > 0$, we have

  $$\mu\|x-y\|^2 \leq \langle \nabla f(x) - \nabla f(y), x-y\rangle \quad \text{for all } x,y \in \mathbb{R}^n \ (f \text{ strongly convex})$$

  $$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x-y\| \quad \text{for all } x,y \in \mathbb{R}^n \text{ and all } i \in [m] \ (f \text{ with Lip-grads})$$

  the gradient method can find the minimum of $f$ with a geometric rate, when stepsize $\alpha$ is suitably chosen
- Implication for the (simple) consensus-based method
- As it cannot used a fixed stepsize, **it cannot achieve a geometric rate!**
- The quest for fast distributed methods has started

# Achieving Geometric Rate: Gradient Tracking

- Lets get back to undirected graph $\mathbb{G} = ([m], \mathcal{E})$

- In weighted-average consensus-based distributed method, the agents were selfish (applies to the push-sum-based method as well)

$$x_i(t+1) = \underbrace{\sum_{j=1}^{m} w_{ij} x_j(t)}_{\text{collaborative}} - \alpha \underbrace{\nabla f_i(x_i(t))}_{\text{selfish}}$$

  where we use a fixed stepsize

- In the models with gradient tracking, the agents are "aware" that there is a system objective and they collaborate on both the decisions and the directions

- **Basic Idea**: In DeGroot consensus model, with $W$ doubly stochastic agent $i$ iterate $x_i(t+1) = \sum_{j=1}^{m} w_{ij} x_j(t)$ tracks the average of the agents' iterates $x_j(t)$, $j \in [m]$

- The iterate $\sum_{j=1}^{m} w_{ij} x_j(t)$ is sufficient to properly track the averages $(1/m) \sum_{j=1}^{m} x_j(t)$ since the agent use no additional information (no other inputs in the system)

- Apply the same idea to gradients: **DIGing – Distributed Inexact Gradient track-ing**
  Each agent uses an estimate $g_i(t)$ to track the gradient averages of all the agents

$$x_i(t+1) = \sum_{j=1}^{m} w_{ij} x_j(t) - \alpha g_i(t)$$

$$g_i(t+1) = \sum_{j=1}^{m} w_{ij} g_j(t) + \underbrace{\nabla f_i(x_i(t+1)) - \nabla f_i(x_i(t))}_{\text{innovation/new input}}$$

- Agents exchange both decision estimates $x_j(t)$ and the gradient estimates $g_j(t)$ with their neighbors

- The updates are reminiscent of "tracking/filtering":
  predicted state + the innovation term

- The innovation term is needed to "track gradients" since the gradient difference is a "new information/new input" to the system from agent $i$.

- Through the exchange of $g_i(t)$ and the consensus step $\sum_{j=1}^{m} w_{ij} g_j(t)$, these local agent inputs (from times prior to $t$) are eventually spread to all agents in the graph

# Closely Related Literature and Simultaneous Work

- Tracking technique used in (not for gradients)
  M. Zhu and S. Martínez, *Discrete-Time Dynamic Average Consensus*, Automatica, 46 (2010),

- A method using gradient tracking proposed in
  J. Xu, S. Zhu, Y. Soh, and L. Xie, *Augmented Distributed Gradient Methods for Multi-Agent Optimization Under Uncoordinated Constant Stepsizes*, in Proceedings of the 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 2055–2060.

- A part of Xu's thesis work
  J. Xu, *Augmented Distributed Optimization for Networked Systems*, PhD thesis, Nanyang Technological University, 2016.

- G. Qu and N. Li, *Harnessing Smoothness to Accelerate Distributed Optimization*, IEEE Transactions on Control of Network Systems 5 (3) 1245–1260, 2018.

# Algorithms NEXT and SONATA

- NEXT by Lorenzo and Scutari - considers general non-convex (objective) problems and a class of algorithms
  P. Di Lorenzo and G. Scutari *Distributed nonconvex optimization over networks*, in IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015, pp. 229–232.
  P. Di Lorenzo and G. Scutari, *NEXT: In-Network Nonconvex Optimization*, IEEE Transactions on Signal and Information Processing over Networks, 2016.
  P. Di Lorenzo and G. Scutari *Distributed nonconvex optimization over time-varying networks*, in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 4124–4128.

- SONATA and its asynchronous variants; convex and nonconvex problems
  Y.Sun, G. Scutari, D.Palomar *Distributed Nonconvex Multiagent Optimization Over Time-Varying Networks* https://arxiv.org/abs/1607.00249, 2016
  Y. Tian, Y. Sun, B. Du, G. Scutari *ASY-SONATA: Achieving Geometric Convergence for Distributed Asynchronous Optimization* Allerton Conference on Communication,

Control, and Computing (Allerton) 2018

Y. Sun, A. Daneshmand, G. Scutari *Convergence Rate of Distributed Optimization Algorithms Based on Gradient Tracking* https://arxiv.org/abs/1905.02637, 2019

- Our work* was motivated by the desire to have a distributed algorithm with a geometric convergence rate†

$$\|x_i(t) - x^*\| \leq q^t M, \qquad \text{for some } M > 0,\ q \in (0,1), \text{ and for all agents } i \in [m].$$

- "Linear" convergence rate is the same as "geometric"

*A.N., A. Olshevsky and W. Shi, Achieving Linear Convergence For Distributed Optimization Over Deterministic Time-Varying Graphs," SIAM Journal on Optimization 27 (4) 2597–2633, 2017

†In fact such a rate is referred to as $R$-linear. When an algorithm is used to solve $\min_{z\in\mathbb{R}^n} f(z)$ and produces iterates $\{z(t)\}$ converging to an optimal solution $z^*$ such that, for some $q \in (0,1)$, we have $\|z(t)-z^*\| \leq q^t\|z(0)-z^*\|$ for all $t \geq 0$, it is said that the method converges with a linear rate to $z^*$. The same terminology is used if the iterates converge in function values, i.e., for some $a \in (0,1)$, we have $0 \leq f(z(t)) - f^* \leq a^t(f(z(0)) - f^*)$ for all $t \geq 0$, where $f^* = \min_{z\in\mathbb{R}^n} f(z)$

# DIGing: Important Gradient Tracking Feature

$$x_i(t+1) = \sum_{j=1}^{m} w_{ij} x_j(t) - \alpha g_i(t)$$

$$g_i(t+1) = \sum_{j=1}^{m} w_{ij} g_j(t) + \nabla f_i(x_i(t+1)) - \nabla f_i(x_i(t))$$

- Initialization: $x_i(0) \in \mathbb{R}^n$ is arbitrary for all $i$, while $g_i(0) = \nabla f_i(x_i(0))$
- $W$ is doubly stochastic
- Consider $\sum_{i=1}^m g_i(t)$:

$$\sum_{i=1}^{m} g_i(t) = \sum_{i=1}^{m} \nabla f_i(x_i(t)) \qquad \text{for all } t \geq 0$$

- The proof is by the induction on the time $t$ and uses **column-stochasticity of** $W$

# Is DIGing Method Correct?

- Will the method solve the scalar problem $\min_{x\in\mathbb{R}} \sum_{i=1}^m f_i(x)$ (chosen for simplicity)?

- Reformulate the problem to capture the agent system

$$\text{minimize} \quad F(x) = \sum_{i=1}^m f_i(x_i) \qquad \text{subject to } x \in \{s\mathbf{1} \mid s \in \mathbb{R}\}$$

where $x = (x_1, \ldots, x_m)'$. Note that $\{s\mathbf{1} \mid s \in \mathbb{R}\} \subset \mathbb{R}^m$ is the consensus subspace.

- DIGing: $W$ is doubly stochastic

$$x_i(t+1) = \sum_{j=1}^m w_{ij} x_j(t) - \alpha g_i(t)$$

$$g_i(t+1) = \sum_{j=1}^m w_{ij} g_j(t) + \nabla f_i(x_i(t+1)) - \nabla f_i(x_i(t))$$

- Define $x(t) = (x_1(t), \ldots, x_m(t))'$ and similarly define $g(t)$

- In a vector form, iterates are given by

$$x(t+1) = Wx(t) - \alpha g(t), \qquad g(t+1) = Wg(t) + \nabla F(x(t+1)) - \nabla F(x(t))$$

- Suppose agents' iterates converge $\lim_{t\to\infty} x(t) = x_\infty$ and $\lim_{t\to\infty} g(t) = g_\infty$.

- In the limit of the method, we have

$$x_\infty = Wx_\infty - \alpha g_\infty, \qquad g_\infty = Wg_\infty$$

- Since $W$ is induced by a connected graph, $g_\infty = Wg_\infty$ implies that $g_\infty = c\mathbf{1}$ for some $c \in \mathbb{R}$.

- Use this in the equation for $x_\infty$: $(I - W)x_\infty = -\alpha c\mathbf{1}$, implying that the vector $\alpha c\mathbf{1}$ lives in the range space of $I - W$.

- The range space of $I - W$ is the same as the subspace orthogonal to the null space of $I - W'$

- The matrix $W$ is doubly stochastic implying that the null space of $I - W'$ is the consensus subspace $S = \{s\mathbf{1} \mid s \in \mathbb{R}\}$.

- Thus, the vector $\alpha c\mathbf{1}$ lives in the subspace $S^{\perp}$. This is possible only for $c = 0$.

- Hence, we must have

$$g_\infty = 0, \qquad (I - W)x_\infty = 0$$

- Implying $x_\infty \in S$ (consensus, all entries are the same): $x_\infty = s^*\mathbf{1}$

- Recall that algorithm has the property $\sum_{i=1}^m g_i(t) = \sum_{i=1}^m \nabla f_i(x_i(t))$ implying that (by taking limits, and using $g_\infty = 0$ and $x_\infty = s^*\mathbf{1}$)

$$0 = \sum_{i=1}^m \nabla f_i(s^*)$$

- Hence $s^* \in \mathbb{R}$ is such that

$$\nabla \left( \sum_{i=1}^m f_i(s^*) \right) = 0$$

which is the optimality condition for a solution of the problem $\min_{x \in \mathbb{R}} \sum_{i=1}^m f_i(x)$

# DIGing Method for Undirected Graphs

- We assume that the graphs are time-varying
- **Exchange**:
  Every agent $i$ sends $x_i(k), g_i(k)$ to all its neighbors $j \in \mathcal{N}_i(k)$ in the graph $\mathcal{G}(k)$ and receives $x_j(k), g_j(k)$ from its neighbors
- **Update**: Every agent $i$ updates the decision and the direction as follows

$$x_i(k+1) = \sum_{j=1}^m W_{ij}(k)x_j(k) - \alpha g_i(k);$$
$$g_i(k+1) = \sum_{j=1}^m W_{ij}(k)g_j(k) + \nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k)).$$

- The (mixing) matrix $W(k)$ is compatible with the graph $\mathbb{G}(k)$:
  $W_{ij}(k) > 0$ for $\{i,j\} \in \mathcal{E}(k)$ and $i = j$, otherwise $W_{ij}(k) = 0$.
- The stepsize $\alpha$ is common to all agents[‡]
- The method is initialized with arbitrary $x_i(0) \in \mathbb{R}^n$ and $g_i(0) = \nabla f_i(x_i(0))$ for all $i$.

[‡]It can be agent dependent AN, Alex Olshevsky, Wei Shi, Cesar Uribe *Geometrically Convergent Distributed Optimization with Uncoordinated Step-Sizes*, CDC 2016.

# Assumptions for Linear Convergence Rate for DIGing

- The functions $f_i$ are convex with Lipschitz continuous gradients with constant $L_i > 0$,

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i \|x - y\| \quad \text{for all } x, y \in \mathbb{R}^n$$

- The average-sum function $\frac{1}{m}\sum_{i=1}^m f_i$ is strongly convex with a constant $\bar{\mu} > 0$

$$\bar{\mu}\|x - y\|^2 \leq \langle \nabla f(x) - \nabla f(y), x - y \rangle \quad \text{for all } x, y \in \mathbb{R}^n$$

- The graphs $\mathbb{G}(k)$ are $B$-connected: for some integer $B \geq 1$, the graph $([m], \cup_{t=k}^{k+B-1} \mathcal{E}(t))$ is connected for all $k$.
- $W(k)$ is doubly stochastic, compatible with the graph $\mathbb{G}(k)$, and there is a $\tau > 0$ such that for all $k$,

$$W_{ij}(k) \geq \tau \qquad \text{whenever} \quad W_{ij}(k) > 0.$$

Under these assumptions we have the following result.

**Theorem 1 (DIGing: Geometric rate)** *The sequences $\{x_i(k)\}, i \in [m]$, generated by DIGing converge to the unique optimal[§] solution $x^*$ at a global R-linear rate $O(\lambda^k)$, where $\lambda \in (0, 1)$ depends on the stepsize $\alpha$, the condition number $\bar{\kappa} = \frac{L}{\bar{\mu}}$ with $L = \max_i L_i$, the connectivity constant $B$, and the mixing matrices (graphs).*

[§] Guaranteed to exist and be unique due to the strong convexity of the objective function

# DIGing: Explicit Geometric Rate

The sequences generated by DIGing converge to the unique optimal solution $x^*$ at a global $R$-linear rate $O(\lambda^k)$, i.e., for some $M > 0$ and $\lambda \in (0,1)$

$$\|x_i(k) - x^*\| \leq \lambda^k M, \qquad \text{for all agents } i \in [m].$$

where, for any step-size $\alpha \in \left(0, \frac{1.5(1-\sigma)^2}{\bar{\mu}C}\right]$, we have

$$\lambda = \begin{cases} \sqrt[2B]{1 - \frac{\alpha\bar{\mu}}{1.5}}, & \text{if } \alpha \in \left(0, \frac{1.5\left(\sqrt{C^2+(1-\sigma^2)C}-\sigma C\right)^2}{\bar{\mu}C(C+1)^2}\right], \\ \sqrt[B]{\sqrt{\frac{\alpha\bar{\mu}C}{1.5}} + \sigma}, & \text{if } \alpha \in \left(\frac{1.5\left(\sqrt{C^2+(1-\sigma^2)C}-\sigma C\right)^2}{\bar{\mu}C(C+1)^2}, \frac{1.5(1-\sigma)^2}{\bar{\mu}C}\right], \end{cases}$$

$$C \triangleq 3\bar{\kappa}B^2\left(1 + 4\sqrt{m}\sqrt{\bar{\kappa}}\right),$$

where $\sigma$ is a uniform upper bound for the spectral radius of matrices $W_k - \frac{1}{m}\mathbf{1}\mathbf{1}'$ for all $k \geq 0$, and $\bar{\kappa} = \frac{L}{\bar{\mu}}$, $L = \max_i L_i$.

# Specialized Result

**Corollary 2 (DIGing: Polynomial networks scalability)** *If the graphs are undirected and each $W(k)$ is a lazy Metropolis matrix (Metropolis-Hastings)*

$$W_{ij}(k) = \begin{cases} 1/\left(1+\max\{d_i(k), d_j(k)\}\right), & \text{if } \{i,j\} \in \mathcal{E}(k), j \neq i \\ 1 - \sum_{\ell \in \mathcal{N}_i(k)} W_{i\ell}(k), & \text{if } j = i, \\ 0, & \text{else}, \end{cases}$$

*where $d_i(k)$ is the degree of a node $i$, and the agents choose a particular step-size $\alpha$*

$$\alpha = \frac{3(2/71)^2}{128B^2m^{4.5}L\sqrt{\bar{\kappa}}} - \frac{1.5}{\bar{\mu}}\left(\frac{(2/71)^2}{128B^2m^{4.5}\bar{\kappa}^{1.5}}\right)^2,$$

*then to reach an $\varepsilon$-accuracy, the number of iterations needed by DIGing algorithm is*

$$O\left(B^3m^{4.5}\bar{\kappa}^{1.5}\ln\frac{1}{\varepsilon}\right).$$

- Analysis uses "small-gain" theorem¶
- We have a variant of DIGing method for directed graphs, relying on push-sum consensus, and its convergence rate is also $R$-linear
- If we used a different analysis (as discussed yesterday), the dependence on $m$ would be in the order of $m$ (for regular graphs - a hunch)
- **A polynomial scaling for a directed graph is still an open question.**

¶AN, A. Olshevsky, W. Shi *Achieving Geometric Convergence for Distributed Optimization over Time-Varying Graphs* SIAM Journal on Optimization 27 (4) 2597–2633, 2017

# Push-Pull Method[‖]

- Works on both undirected and directed graphs, but **static** i.e., $\mathbb{G} = ([m], \mathcal{E})$.
- It is a variant of DIGing that uses different matrices for mixing the decisions and the directions
- **Exchange**: (from an agent's perspective
  - (**Pull**) Every agent $i$ receives $x_j(k) - \alpha g_j(k)$ from its in-neighbors $j \in N_i^{\text{in}}$
  - (**Push**) Every agent $i$ sends $C_{\ell i} g_i(k)$ to all its out-neighbors $\ell \in N_i^{\text{out}}$
- **Update**: Every agent $i$ updates its decision $x$ and direction $g$ as follows

$$x_i(k+1) = \sum_{j=1}^m R_{ij} \left(x_j(k) - \alpha g_j(k)\right);$$
$$g_i(k+1) = \sum_{j=1}^m C_{ij} g_j(k) + \nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k)).$$

The matrix $R$ is row-stochastic, while $C$ is a column stochastic!!!

$r_{ij} = 0$ if $j \notin \mathcal{N}_i^{\text{in}}$ and $c_{ij} = 0$ if $j \notin \mathcal{N}_i^{\text{out}}$.

The method is initialized with arbitrary $x_i(0) \in \mathbb{R}^n$ and $g_i(0) = \nabla f_i(x_i(0))$ for all $i$.

The stepsize $\alpha$ can be agent dependent.

[‖] S. Pu, W. Shi, J. Xu, A. N. "Push-Pull Gradient Methods for Distributed Optimization in Networks," IEEE TAC 2021

# Alternative

Method

- (**Pull**) Every agent $i$ receives $x_j(k) - \alpha g_j(k)$ from its in-neighbors $j \in N_i^{\text{in}}$
- (**Push**) Every agent $i$ sends $C_{\ell i} g_i(k)$ to all its out-neighbors $\ell \in N_i^{\text{out}}$

$$x_i(k+1) = \sum_{j=1}^m R_{ij}\,(x_j(k) - \alpha g_j(k));$$
$$g_i(k+1) = \sum_{j=1}^m C_{ij} g_j(k) + \nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k)).$$

**A variant that is simpler but not as safe for privacy**

- (**Pull**) Every agent $i$ receives $x_j(k)$ from its in-neighbors $j \in N_i^{\text{in}}$
- (**Push**) Every agent $i$ sends $C_{\ell i} g_i(k)$ to all its out-neighbors $\ell \in N_i^{\text{out}}$

$$x_i(k+1) = \sum_{j=1}^m R_{ij} x_j(k) - \alpha g_i(k);$$
$$g_i(k+1) = \sum_{j=1}^m C_{ij}(k) g_j(k) + \nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k)).$$

Each of these methods have the Gradient Tracking Feature (under the given initialization) and column-stochasticity of $C$:

$$\sum_{i=1}^m g_i(t) = \sum_{i=1}^m \nabla f_i(x_i(t)) \qquad \text{for all } t \geq 0$$

# Simultaneous Work

- S. Pu, W. Shi, J. Xu, and A. Nedić, *A push-pull gradient method for distributed optimization in networks*, Proceedings of the 54th IEEE Conference on Decision and Control (CDC), 2018; journal version on arxiv: https://arxiv.org/abs/1810.06653

- C. Xi, V. S. Mai, R. Xin, E. H. Abed, and U. A. Khan, *Linear convergence in optimization over directed graphs with row-stochastic matrices*, IEEE Transactions on Automatic Control, 2018.

- R. Xin, C. Xi, and U. A. Khan, *Frost–fast row-stochastic optimization with uncoordinated step-sizes*, EURASIP Journal on Advances in Signal Processing, 2019.

- R. Xin and U. A. Khan, *A linear algorithm for optimization over directed graphs with geometric convergence*, arXiv preprint arXiv:1803.02503, 2018; IEEE Control Systems Letters 2 (3) 315 – 320, 2018.

# Interpretation of Push-Pull Method

- Suppose the graph $\mathbb{G} = ([m], \mathcal{E})$ is strongly connected, and suppose $x \in \mathbb{R}$ (for simplicity)

- Reformulate the problem to capture the agent system

$$\text{minimize} \quad F(x) = \sum_{i=1}^{m} f_i(x_i) \qquad \text{subject to } x \in \{s\mathbf{1} \mid s \in \mathbb{R}\}$$

  where $x = (x_1, \ldots, x_m)'$.

- Define $x(k) = (x_1(k), \ldots, x_m(k))'$ and define $g(k)$, similarly

- In this compact representation the Push-Pull iterations are

$$x(k+1) = R(x(k) - \alpha g(k))$$
$$g(k+1) = Cg(k) + \nabla F(x(k+1)) - \nabla F(x(k))$$

If the sequences were convergent, i.e., $x(k) \to \bar{x}$ and $g(k) \to \bar{\mathbf{g}}$, then we have

$$\bar{\mathbf{x}} = R(\bar{\mathbf{x}} - \alpha\bar{\mathbf{g}}), \qquad \bar{\mathbf{g}} = C\bar{\mathbf{g}}.$$

Hence

$$(I - R)\bar{\mathbf{x}} = -\alpha R\bar{\mathbf{g}}, \qquad (I - C)\bar{\mathbf{g}} = 0.$$

- The last equation implies that $\bar{\mathbf{g}}$ is the null space of $I - C$. Since the graph is connected, it follows that $\bar{\mathbf{g}} = s\pi_c$ for some $s \in \mathbb{R}$, where $\pi_c > 0$ is the Perron vector for $C$.

- Using $\bar{\mathbf{g}} = s\pi_c$ in the first relation and the fact that $R$ is row-stochastic with least two positive entries we find that

$$R\pi_c = v \qquad \text{with } v > 0$$

and obtain

$$(I - R)\bar{\mathbf{x}} = -\alpha s R\pi_c = -\alpha s v \qquad \text{with } v > 0$$

- Thus, the vector $-\alpha s v$ lies in the range space of $I - R$.

- The range space of $I - R$ is the same as the space $(\text{null}(I - R'))^{\perp}$, implying that the vector $-\alpha s v$ is orthogonal to the null space of $I - R'$.

- The null space of $I - R'$ is spanned by the Perron vector $\pi_r$, implying that $-\alpha s v$ is orthogonal to $\pi_r$,i.e.,

$$-\alpha s \langle v, \pi_r \rangle = 0$$

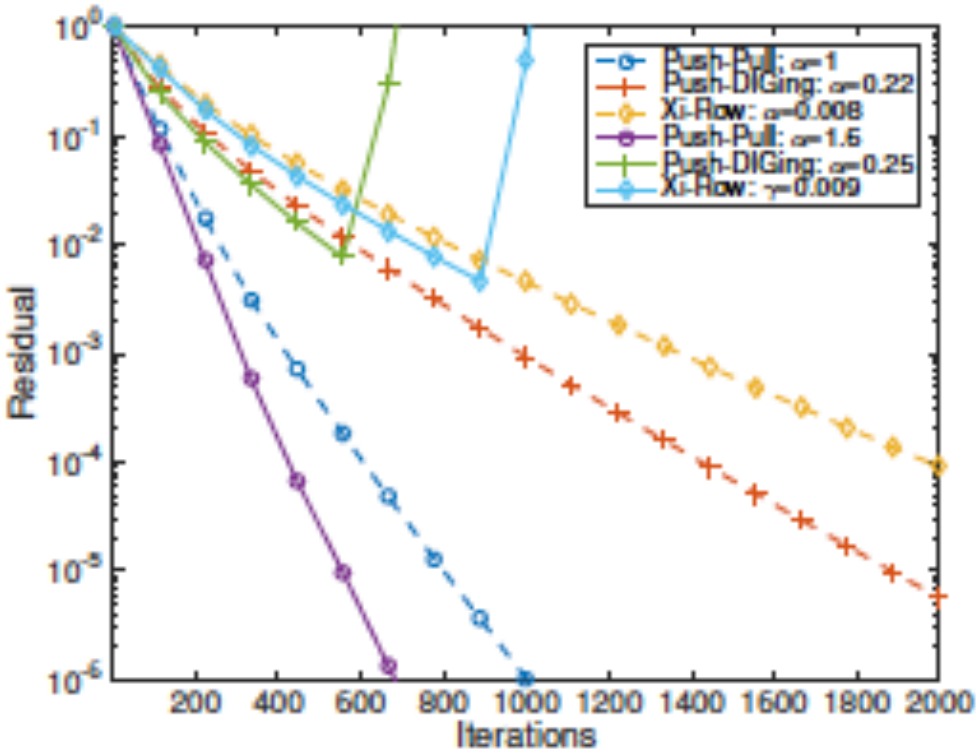
- Since stepsize $\alpha > 0$, and $v > 0$ and $\pi_r > 0$, the preceding relation can hold only for $s = 0$. Hence

$$\bar{\mathbf{g}} = 0$$

Thus the relation for $\bar{\mathbf{x}}$ reduces to $(I - R)\bar{\mathbf{x}} = 0$.

- From now on, we argue along the same lines as in DIGing to conclude that $\bar{\mathbf{x}} = s^* \mathbf{1}$, and $s^*$ is the solution of the problem.

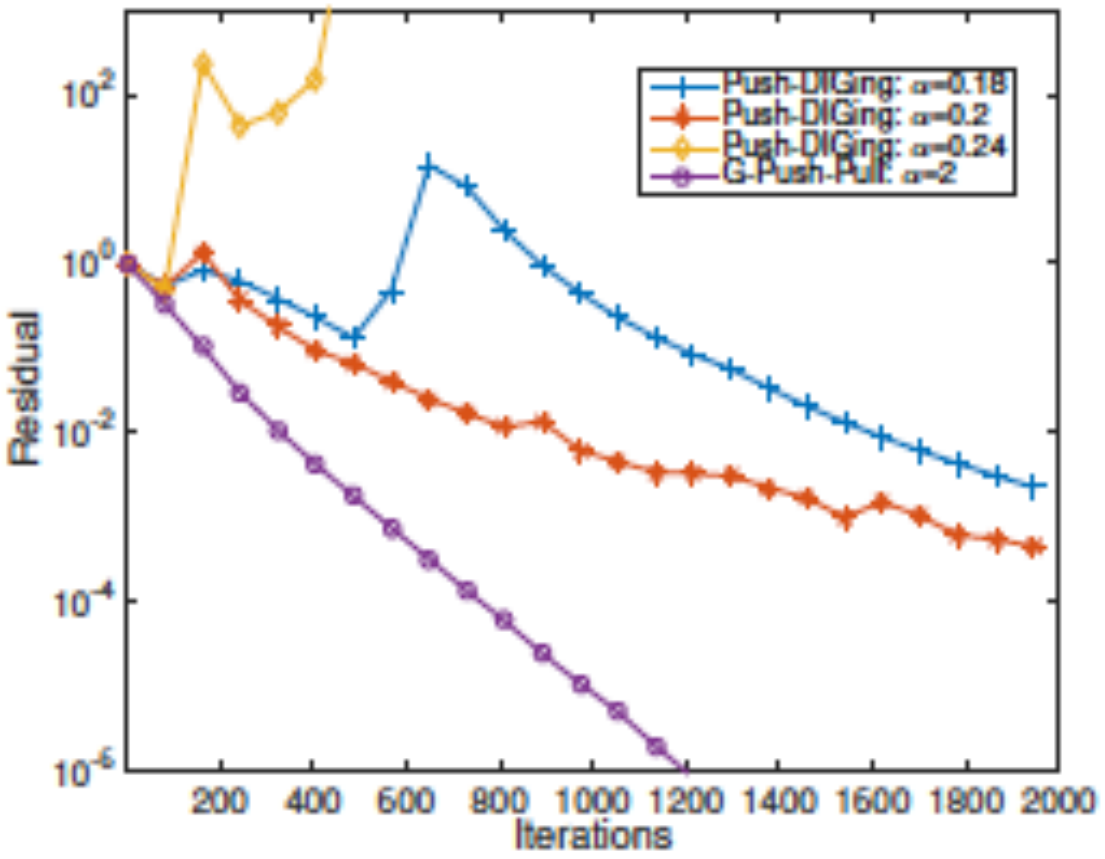
# Simulations: Push-Pull




(a) Fixed directed network

- The objective functions are quadratic. The network is random with 20 nodes
- The plot shows the normalized residuals $\sum_{i=1}^m \|x_i(k) - x^*\|^2 / \sum_{i=1}^m \|x_i(0) - x^*\|^2$.

# Simulations: Gossip-like Push-Pull




(b) Asynchronous directed network

- The same setting, with a random agent that wakes up.

- It selects two sets of random neighbors, one for decision updates, and the other for direction update.

- The plot shows the normalized residuals $\sum_{i=1}^m \|x_i(k) - x^*\|^2 / \sum_{i=1}^m \|x_i(0) - x^*\|^2$ averaged over 20 runs.

# Convergence Result

Assume that:

- The graph $\mathbb{G}$ is directed and strongly connected
- The matrices $R$ and $C$ are compatible with the graph and, respectively, row-stochastic and column-stochastic
- Each $f_i$ has Lipschitz continuous gradients with a constant $L > 0$
- The sum $\sum_{i=1}^{m} f_i$ is strongly convex with a constant $\mu > 0$

**Proposition 3** *Under these assumptions the Push-Pull Method produces the iterate sequences* $\{x_i(t)\}$ *that converge geometrically fast to the optimal solution of the problem for a sufficiently small stepsize* $\alpha$.

The analysis makes use of the Perron vectors (probability vectors) for $R$ and $C$:

$$\pi_r' R = \pi_r', \qquad C\pi_c = \pi_c,$$

and the weighted average of $x_1(k), \dots, x_m(k)$.

$$\bar{x}(k) = \pi_r' \mathbf{x}(k),$$

where $\mathbf{x}(k)$ is the matrix with rows $x_i(k)'$, $i \in [m]$.

The progress of the algorithm is measured in terms of three quantities:

$$\|\bar{x}(k) - x^*\|, \qquad \|\mathbf{x}(k) - \mathbf{1}\bar{x}(k)\|_{\pi r} = \left(\sum_{i=1}^m [\pi_r]_i \|x_i(k) - \bar{x}'(k)\|^2\right)^{1/2},$$

$$\|\mathbf{g}(k) - \pi_c s(k)'\|_{\pi_c^{-1}} = \left(\sum_{i=1}^m \frac{\|g_i(k) - s(k)[\pi_c]_i\|^2}{[\pi_c]_i}\right)^{1/2}, \qquad \text{with } s(k) = \sum_{i=1}^m g_i(k)$$

Main relation:

$$\begin{bmatrix} \|\bar{x}(k+1)-x^*\|^2 \\ \|\mathbf{x}(k+1)-\mathbf{1}\bar{x}(k+1)\|^2_{\pi_r} \\ \|\mathbf{g}(k+1)-\pi_c s(k+1)'\|^2_{\pi_c^{-1}} \end{bmatrix} \leq D(\alpha) \begin{bmatrix} \|\bar{x}(k)-x^*\|^2 \\ \|\mathbf{x}(k)-\mathbf{1}\bar{x}(k)\|^2_{\pi_r} \\ \|\mathbf{g}(k)-\pi_c s(k)'\|^2_{\pi_c^{-1}} \end{bmatrix}$$

$$D(\alpha) = \begin{bmatrix} 1-O(\alpha) & O(\alpha) & O(\alpha^2) \\ O(1) & 1-\sigma_2(R) & O(\alpha^2) \\ O(1) & O(1) & 1-\sigma_2(C)+O(\alpha) \end{bmatrix}$$

- $\sigma_2(A)$ is the second largest singular value of a matrix $A$

Then, the all three quantities converge to 0 at a linear rate with coefficient $\rho_{D(\alpha)} < 1$, where $\rho_D$ is a spectral radius of a matrix $D$, provided that the stepsize is small enough.

- Details and some small scale simulations can be found in:
  S. Pu, W. Shi, J. Xu, and AN *Push-pull gradient methods for distributed optimization in networks*, 2018, arXiv preprint at https://arxiv.org/abs/1810.06653. IEEE TAC 2021.
- Closely related recent paper:
  R. Xin, A.K. Sahu, U.A. Khan, and S. Kar *Distributed stochastic optimization with gradient tracking over strongly connected networks* CDC 2019

# Conclusion

- Fast distributed gradient methods are developed that can match the best performance of centralized gradient methods
- New directions
  - Solving nonconvex problems (T. Tatarenko & B. Touri 2017, A. Scutari's group at Purdue)
  - Asynchronous implementations (S. Pu, A. Scutari's group)
  - Impact of network topology (N. Neglia at INRIA, A. Olshevsky at BU)
  - Impact of delays (M. Johansson at KTH, M.G. Rabbat at Facebook/McGill)
  - Performance in presence of malicious agents (S. Sundaram, N. Vaidya, A. Scaglione, W.U. Bajwa, AN)
  - Privacy (Y. Wang at Clemson University)